NASA Technical Memorandum 88372

# On the Use of Composite Grid Schemes in Computational Aerodyanmics

Joseph L. Steger and John A. Benek

November 1986

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 88372

# On the Use of Composite Grid Schemes in Computational Aerodyanmics

Joseph L. Steger, Ames Research Center, Moffett Field, California
John A. Benek, Calspan Corp., AEDC, Arnold AFS, Tennessee

November 1986

National Aeronautics and Space Administration

**Ames Research Center**
Moffett Field, California 94035

# ON THE USE OF COMPOSITE GRID SCHEMES IN COMPUTATIONAL AERODYNAMICS

Joseph L. Steger and John A. Benek

ABSTRACT

In finite difference flow field simulations, the use of a single well-ordered body conforming curvilinear mesh can lead to efficient solution procedures. However, it is generally impractical to build a single grid of this type for complex three dimensional aircraft configurations. As a result, a trend in computational aerodynamics has been toward the use of composite grids. Composite grids use more than one grid to mesh an overall configuration with each individual subgrid of the system patched or overset together. Because each individual subgrid in the system is well-ordered, the overall grid is suitable for efficient finite difference solution using vectorized or multitasking computers. Some of the advantages and difficulties of using various composite grid schemes are reviewed in this paper.

## INTRODUCTION

Over the last decades, efficient and sophisticated finite difference schemes have evolved for treating complex flow fields about relatively simple aerodynamic configurations by generally using curvilinear body conforming grids. These procedures are now being extended to more complex shapes through the use of composite grid schemes. In such a domain decomposition scheme the field is meshed by using a set of interconnected simple grids that make up a larger composite grid. Grids are either patched or overset together to form a composite grid capable of treating complex geometries.

In some ways a composite mesh finite difference scheme assumes some of the characteristics of a finite element scheme that uses a few large powerful elements in which each element is itself discretized. This process is still in its infancy, and not all of the ways of connecting the composite grids have been explored. In this paper we will briefly review some of the composite grid approaches used in aerodynamic applications.

## BACKGROUND

Aerodynamic simulation has been undertaken in order to provide understanding of complex flows and to provide design inputs. However, the simulation of aerodynamic flows has been challenging because of two phenomena: multiple length scales and nonlinearity. In most aerodynamic flows the Reynolds number is quite high and viscosity is effective only in a small region near the wall where it moderates the no-slip boundary condition. This feature alone induces two length scales, a thin inner viscous region which must be resolved along with an outer essentially inviscid flow region. Consequently, two approaches to simulating aerodynamic flows have evolved - either solve the Euler and Navier Stokes equations everywhere (in which case the grids must be constructed to resolve the viscous layer), or solve special approximate equations in regions where they are valid, such as boundary layer and potential flow equations. The other challenging aspect, equation nonlinearity, characterizes shock waves, vortex breakdown and burst, buffet, buzz, and stalling. Because of such phenomena, even flow about a relatively simple configuration can be so complex that its simulation remains a major objective of computational aerodynamics.

Finite difference methods have been developed to solve Euler and Navier-Stokes equations as well as potential and boundary layer equations. These methods have proven to be quite satisfactory insofar that wave propagation, viscous layers, shock waves, and unsteady motions can all be treated. The finite difference method is flexible and previous developments in aerodynamic theory such as perturbation analysis and various approximate techniques can be incorporated into its use. Because relatively simple configurations can cause interesting physical effects, the emphasis has been on their simulation. For a simple configuration, a finite difference simulation can often be carried out on a single well-ordered grid that can be body-conforming as well as clustered to flow field action regions. This is advantageous because such meshes allow the use of efficient numerical solution procedures and computer architectures. For example, on a well-ordered mesh implicit alternating direction methods can be used to approximately invert large sparse matrices and these methods readily adapt to vectorized computer processors.

The prediction of flow about more complex aerodynamic configurations is currently carried out using very simplified governing flow field equations or boundary condition treatments. For example, panel methods [1,2] for predicting the flow about highly complex geometries solve linear potential flow equations. In this approach, viscous and vortical effects are neglected or modeled and are included as boundary conditions. Flow about complex configurations has also been simulated using finite difference schemes for transonic small-disturbance potential equations and boundary conditions [3,4]. But as finite difference procedures for solving the more exact equations become more reliable, efficient, and versatile, and as computing becomes more cost effective, there is an interest in extending these more exact procedures toward complex geometries.

In Fig. 1 are shown three basic grid treatments for meshing a simple body – a rectangular or Cartesian-like grid, a curvilinear body-conforming grid, and an irregular triangularized grid. Each grid type has advantages and disadvantages. The rectangular grid is well-ordered, trivial to generate, readily allows accurate interior difference approximations, and the representation of a difference approximation requires the minimum work per step. However, boundary representation requires special logic and is generally of poor accuracy, and the grid does not cluster to efficiently resolve viscous boundary layers on curved boundaries. The curvilinear body-conforming mesh is also well-ordered, allows higher order difference approximations, permits simple and accurate boundary difference approximations, and can be clustered to gradient regions. It is especially well suited for viscous boundary layer approximation. However, the governing equations are more complex to difference on a curvilinear grid (although body conforming grids often permit use of additional approximations), and grid generation, while not difficult for simple bodies, is no longer trivial. The triangularized mesh has good grid concentration (i.e. triangles can be readily deleted in smooth gradient regions) and the shape of the boundary curve is readily conformed to. However, such a mesh is poorly ordered and is therefore less amenable to the use of certain algorithms (e.g. ADI) and vectorized computers. Grid generation can be relatively straightforward, but is not trivial, and high order accuracy appears more difficult to maintain unless direct inversion routines are used to solve the difference approximations. Moreover, triangular meshes are a poor choice for resolving viscous boundary layers in two dimensions because either highly acute or too many triangles are required. (In three dimensions a grid formed by triangular discretization along the surface and by lines radiating outward from the surface would nicely resolve viscous boundary layers, see Fig. 2. Such grids appear to have good generality and are "semi well-ordered" for reasonable efficiency.)

For a simple body shape, the use of a single body-conforming curvilinear mesh leads to the most efficient solution procedure. However, for more complex shapes, the generation of a body conforming well-clustered curvilinear grid that is not overly skewed and has smooth variation can often be quite difficult. In particular, it is generally impractical to build a single grid of this type for complex three dimensional configurations. Of course, by judiciously introducing cuts in the grid, some fairly complex bodies can be meshed with a single grid; the sketch shown in Fig. 3 illustrates this possiblility. Single grids can also be quite effective for certain complex configurations in which the body shape has a smooth variation and the main flow is supersonic. For supersonic outflow, a simple outflow boundary condition can be used, allowing use of a grid chopped at the back. Cross sections of such a grid that was generated for space shuttle simulations are shown in Fig. 4.

To treat complex geometries by the finite difference (or finite element) method, a code builder can: 1) revert to grid systems such as the irregular triangularized mesh systems, 2) form a composite grid by patching or oversetting well-ordered grids together, or, 3) use a composite grid in which only some of the grid segments are well-ordered. This paper will only discuss aspects of composite grids in which the overall configuration is meshed by either patching or oversetting well-ordered grids together.

## COMPOSITE GRID SCHEMES

Composite grids use more than one grid to mesh an overall configuration with each individual grid of the system patched together or overset. The sketches shown in Fig. 5 illustrate several simple patched and overset grid configurations in two dimensions for a typical two body problem. As the sketches illustrate, patched grids are individual meshes that are joined together at some common interface surface. With overset grids the meshes are simply superimposed or partially superimposed to cover the region of interest, and are not joined together in any special way, although they can be.

The use of a set of patched or overset grids to form a larger composite grid carries the discretization process one step further. In a sense a composite mesh finite difference process assumes some of the characteristics of a finite element scheme that for the most part uses a few large powerful elements in which each element is itself discretized. In this discretization process each, or nearly each, individual grid in the system is well-ordered and is thus suitable for efficient finite difference solution using vectorized computers and any available single grid code. (The small patched triangularized segment shown in Fig. 5 is not well-ordered.) The problem with a composite grid scheme is the difficulty of accounting for all the possible communications between meshes and the difficulty of supplying interface boundary data without degrading numerical accuracy or convergence.

Limited experience with both patched and overset grids has not shown which method is preferable. An optimum method will likely combine both patched and overset grids and perhaps small grid segments that are not well-ordered, and such grids have already been tried [5,6]. Both patched and overset grid schemes necessitate extensive bookkeeping procedures. One of the drawbacks of the patched grid method is a grid generation problem that is still relatively difficult because various interfaces have to be defined and grids with both inner and outer boundary surfaces must be generated. Drawbacks to overset grids include interpolation of data points along an irregular boundary and bookkeeping which can be especially complex if more than two levels of overset grids intersect each other.

With composite grids the possibility exists of using different computers to update the results on each grid if a multitasking computer processor is available, but this is feasible only if the amount of work on each grid or subdivided grid is roughly the same [7]. Likewise, on machines which have a small but high speed memory and a large but low speed peripheral memory it is feasible to keep only the memory requirements for a given grid internal to the small fast memory. Of course, one can partition a single grid scheme for multitasking and memory can be rolled back and forth, but with composite grid schemes this data management is naturally built into the code.

To help clarify some of the following discussion, Figs. 6 - 7 show published inviscid flow results obtained using patched and overset meshes in two dimensions. Both grids and flow contour levels are shown. The supersonic biplane results of Hessenius and Rai [8] were computed using a patched grid method in which a flux balance interface scheme is used at grid boundaries to ensure flow conservation. Although in this method meshes must share a common boundary, grid lines need not have a common slope or join together. The transonic inviscid flow results of Dougherty et al [9] for an airfoil and flap were computed using overset grids. Only simple interpolation is used to interface the mesh boundaries, taking boundary value data from the underlying grid. Figure 7c shows the nearby interpolation points used to update the outer boundary of the flap grid (open symbols) and grid points which are excluded from the calculation (filled symbols).

## 1) Patched Grid Schemes

A patched grid is perhaps the most natural composite grid approach. The domain is decomposed into a finite number of subdomains or elements and each element is meshed separately. The grids thus share common boundary interfaces. Such an approach can be used to construct simpler or more efficient grids, to treat complex geometries, or to better resolve flow changes.

Compared to a single mesh code about a simple configuration, a patched grid code differs in three major ways: 1) the grid generation process must be carried out for a partitioned grid, 2) multiple grids must be interfaced and managed, and 3) additional numerical stability and accuracy requirements are encountered at subgrid interface boundaries.

On a patched grid, the overall domain is subdivided and a mesh must be supplied for each subgrid. Because the partitioning results in common boundaries, each subgrid must be generated subject to specified boundary constraints. In generating a subgrid the user must specify either the interface boundary surface grid point distribution, or some suitable constraints that restrict boundary grid points to the interface. Care must be taken in subdividing into subgrids so that a specified boundary surface distribution on one face does not cause poorly clustered or highly skewed grid lines to result on another boundary surface.

Once the domain is partitioned using subgrids, a means of bookkeeping and data management must be supplied to indicate how a subgrid fares relative to its neighbors and other boundaries. At any given subgrid boundary the possibility exists that this boundary is in common with another subgrid interface or that it is a body boundary, a free stream boundary, an inflow boundary, etc., or even some combination of these. All of these possibilities must be allowed in the bookkeeping scheme. It is frequently the case that not all of the grids fit into main memory, so a means of collecting needed interface boundary data that must be supplied from another subgrid no longer in memory must also be provided.

One way to supply interface boundary data is by interpolating the interior solution from the grids to either side. Because the subgrids likely join discontinuously as sketched in Fig. 8a, the interpolation process must be constructed with this in mind. One approach is to extrapolate for interface values from the grids to either side and to then average these values to obtain the interface boundary values. The averaging helps maintain numerical stability and permits signal propagation in both directions. Thus a value at point $a$ in the sketch might be obtained by averaging the value obtained by extrapolating from the interior left grid with a value interpolated from points $b$ and $c$ which are extrapolated from the right side grid. An improvement on this which avoids extrapolation and uses the governing equations is to have the grids overlap one point as sketched in Fig. 8b (this is as much an overset grid as it is a patched grid). However, this grid is much more difficult to generate as there are now two common interface surfaces. As Fig.8b helps show, the left grid boundary can now be fully supplied by interpolating only interior values determined from difference equations along a line of the right-side grid. The right-side grid boundary can be obtained in similar fashion.

Characteristic compatibility relations can also be used to provide interface boundary values. For a system of hyperbolic equations of the form

$$\partial_t Q + A\partial_x Q + B\partial_y Q = 0$$

the eigenvalues and eigenvectors of $A$ or $B$ can be used to obtain combinations of variables which can be properly differenced to one side of an interface boundary. For example, in dealing with a boundary in $x$, the eigenvectors $X^{-1}$ of $A$ can be used to partition the equations as

$$X^{-1}\partial_t Q + \Lambda X^{-1}\partial_x Q + X^{-1}BXX^{-1}\partial_y Q = 0$$

where $\Lambda$ is a diagonal matrix containing the eigenvalues of $A$. On the left boundary, the portion of $X^{-1}\partial_x Q$ with negative eigenvalues as coefficients can be differenced to the right (i.e. forward differenced), while on a right boundary the portion of $X^{-1}\partial_x Q$ with positive eigenvalues as coefficients can be differenced to the left (i.e. backward differenced). The remaining portion of the characteristic combination of variables, $X^{-1}Q$, must be supplied from the neighboring grid - either by interpolation or by characteristic relations from that side.

Conservatively differenced flow equations will lose this property at the interface boundary if interpolation or characteristic differencing supplies the interface boundary values. Depending on the location of the interface boundary this may not be a problem, but, if a captured shock wave passes through the boundary, some loss of accuracy may be expected. For this reason special interface boundary conditions have been developed so that flux balancing is maintained between grids ( see the review by M. M. Rai [10]).

For the most part it is difficult to treat the interface boundary updating procedure in a fully implicit way. For example, if the characteristic compatibility relations are used to update the interface values, the number of one-sided difference equations with proper eigenvalues could be implicitly updated with the interior update of the grid. The interface values that depend on the adjacent grid, however, are most likely going to be lagged in iteration or time. As a result of lagging these values, the stability properties of any implicit interior differencing scheme will be degraded somewhat, perhaps enough to require an iterative correction process for the interface boundary values.

As described in Refs. [10-19], patched grids have been used to resolve gradients, treat moving boundaries, and treat complex geometries. As an example of the progress achieved with patched grids, recently obtained results about a moving rotor-stator combination are reproduced from Ref.[19]. The two dimensional inviscid flow calculation used the two simple grids shown in Fig. 9. Supersonic inflow boundaries were specified on the left boundary, and, to represent a cascade row, periodicity conditions were imposed on the lower and upper boundaries. Figures 10a - 10c show pressure contours for the rotor-stator at $M_\infty = 1.5$. The flow is observed from the forward plunging rotor which therefore appears to be stationary while the stator appears to be moving. An effective angle of attack is induced by the downward plunge equal to a Mach number of 0.1, and a cyclic time-dependent flow is induced on the stator row.

## 2) Overset Grid Schemes

In the overset mesh technique individual grids are superimposed to form a composite grid. Usually there is a major grid which might be a simple rectangular grid stretched over the entire field or a grid generated about a dominant boundary or body surface. Minor grids are generated about remaining portions of the body-configuration or any other special feature such as an intense vorticity region. The minor grids are used to resolve features of the geometry or flow that are not adequately resolved by the major grid. They are generated somewhat independently of the major grid and are overset on the major grid without requiring mesh boundaries to join in any special way. (The major grid could itself be a composite grid.)

The two dimensional airfoil-flap combination shown in figure 11 illustrates a typical overset grid scheme. Body conforming meshes are independently generated about both the main airfoil and the flap. The grid generated for the flap is a minor grid and so it is not carried out to the far field (thus the mesh generation is not entirely independent). The two grids are not joined at a common boundary and so are connected by interpolating data from one grid to another. Figure 12 illustrates the typical way in which data are transferred from one grid to another. Here data from the major grid are interpolated to supply outer boundary data to the minor flap grid. (If a characteristic outflow differencing scheme is used at the outer boundary, then only unknown characteristic data must be interpolated.) Hence the presence of the main airfoil is properly imposed on the flap mesh. In order for the major grid solution to account for the flap, points of the major grid are blanked out within some neighborhood of the flap. On the fringes of this blanked-out region, data from the minor grid solution are interpolated to serve as interior boundary data for the major grid. Thus, the effect of the flap is imposed upon the main airfoil. Overall the effect of one grid is imposed upon the other by interpolating boundary data between them, and often this process is carried on as an iterative solution process proceeds on each grid.

With the exception of some additional data management, the only changes required to the flow solver account for holes or interior boundaries in the grid and account for boundary data transferred from another grid rather than from the usual boundary condition routines. For example, for the simple airfoil-flap problem all of the minor grid outer boundary data is interpolated from the major grid solution, and not from the usual far field boundary condition routine. In time accurate or time-like iterative solution routines, accounting for a hole can be very straightforward. The simplest approach is to simply sweep through the entire grid, ignoring the hole, and to use a flag that zeroes the $\Delta t$ or relaxation parameter at a hole or hole-boundary point. Thus no update of the variables takes place within a hole, but the later hole boundary update process ensures that the regular interior grid points sense the result of the other grid solution. Overall the actual flow solver routine is not appreciably complicated by using overset grids.

The data management process and the best ways of doing the interpolation procedures are more involved and have not yet been extensively researched. As far as the interpolation procedure is concerned, issues of efficiency, accuracy and stability remain. The interpolation of grid boundaries would generally be undertaken as an explicit operation at the end of each time step or iterative update for interior values of the grid. Depending on the grid, certain fringes points on a boundary may be updated more by an extrapolation process than by an interpolation process and this can cause numerical instability or slow the iterative convergence. However, if sufficient grid overlap exists, a simple interpolation process usually results in a stable procedure. Finally, interpolation procedures will not generally ensure conservation of flux quantities, and inaccuracy in shock capturing may or may not occur depending on the location of the updated boundaries and the change of spatial resolution across the boundary.

There are other complications and ramifications to using overset grids. For example, as shown in Fig. 7 the outer boundary of the flap grid also intersects the airfoil, so additional logic is needed to put a hole in the minor grid. This presents no additional problems to the flow solver, but the bookkeeping or data management becomes somewhat more involved, and the automatic routine that identifies and manages the grid interconnections becomes more complex. It is also clear that as these two bodies are drawn closer and closer together the grid spacing must shrink to resolve a small gap - so one grid cannot be generated truly independent of another. (In this regard, the mesh spacings should be kept compatible so that the interpolation process is accurate.)

Because overset grids may have common regions of overlap, alternate ways of transferring data are possible. In the procedure just described, the transfer of information from one grid to another is by means of boundary conditions (outer boundary and hole). This is computationally efficient insofar that only a small subset of data is involved in this process. However, because the grids generally overlap, the effect of one grid solution on the other could also be imposed over a region by means of a forcing function. For example, an equation of the type

$$\partial_t Q + L(Q) = 0$$

with $L(Q)$ a spatial differencing operator, can be altered with a forcing function

$$\partial_t Q + L(Q) = \alpha(Q^o - Q)$$

where $Q^o$ represents $Q$ interpolated from an overset grid. In regions where the grids overlap, the value $Q$ on the present grid and the value of $Q^o$ of the overset grid are both available. In these regions $\alpha$ can be chosen greater than 0 and $Q^o$ can be interpolated to the same grid point location as $Q$ to be used in the forcing function. If $Q^o$ represents an inaccurate result or if the grids do not overlap, $\alpha$ is set to 0.

Overset grids have been used in flow applications [20-30] to save grid points, to resolve gradients, to treat complex geometries, and to treat moving boundaries. Recently obtained results about a generic wing-body-tail [30] and a two dimensional unsteady separation problem [9] are shown in Figs 13-14 to illustrate the progress with overset grids.

The wing-body-tail configuration consists of an ogive-cylinder fuselage and untapered swept wings. A main hemispherical-like grid was placed around the fuselage and minor cylindrical-like grids were placed about the wing and the tail. Figure 13a shows the configuration, surface grid point distribution, and outer boundaries of the wing and tail minor grids. Another outermost grid was used to resolve the solution from the near field to either the far field or to wind tunnel wall boundaries. Comparisons of computed results and experimental data are shown in Fig. 13b for a free stream Mach number of 0.9 and an angle of attack of 2 degrees.

The grids and computed Mach contours for the two body separation case are shown in Fig. 14. Here the small body with its minor grid pitches while it translates upward in time. The frames of Fig. 14 show Mach contours and grid positions at two times during the unsteady motion. As the bodies move with respect to each other, shock waves form and are transmitted through the overset grid boundaries without apparent difficulty.

## COMPUTATIONAL ASPECTS

Compared to single mesh codes for aerodynamic simulation, composite mesh computer codes contain considerably more computationally related instructions. Specifically much of the code entails data management features, pointer arrays, grid interfacing routines and so on.

The success of the composite mesh code can depend as much on what can be termed computational geometry features as it depends on the algorithm used to solve the flow field equations on a given grid. For example, in both patched grid schemes and overset grid schemes interpolation is generally needed to update boundary points, and interpolation requires routines to seek out nearest points (or cells) from which to interpolate. Especially in the case of overset grids, finding these points can be costly and prone to error. Every point on the surface or field can be tested to find the one nearest the point to be interpolated, but this near-point algorithm is computationally expensive if the data bases are large. Morever, as Fig. 15 indicates, for fine skewed grids the nearest point (point $a$) is not necessarily the best point to use for interpolation. For the grid segment shown, the point $b$ (or cell $bcde$) is best for interpolation.

As an example of one of the computational procedures that must be developed to use overset grids, we illustrate the steps taken to construct a hole in a grid which is caused by a body overset onto another grid. In general it is necessary to identify and flag all points that lie within the body boundary, say the $\varsigma = 0$ grid surface, so that they can be excluded from the flow field calculation. To ensure that there is no ambiguity, a surface containing the body boundary can be used to define a hole, and, for body-conforming coordinates, this surface can be taken as any level surface above the body boundary, say $\varsigma = 2\Delta\varsigma$ (see Fig. 16). The following algorithm was developed to locate hole points within this boundary (Figs. 16a to 16d help define the notation described below).

### *Hole Point Algorithm*

*Setup:* 1) Define the boundary surface, $C$, which will define the hole. This surface may be defined by all the grid points of the chosen $\varsigma$-level surface, or for economy reasons, some subset of these points, perhaps every other one. 2) Construct outward normals, $\vec{N}$, at each point defining the $\varsigma$-level surface, $C$. 3) Determine a temporary origin within the $\varsigma$-level surface, say $P_0$, which may be located within the surface by simply averaging the points defining $C$. 4) Define a ball about the surface whose

radius $R_{max}$ is the maximum distance from $P_0$ to any point on $C$.
*Test:* 5) Test the magnitude of $\vec{r}$, the position vector from $P_0$ to any point $P$ of the grid being tested. If $|\vec{r}| \geq R_{max}$, the point $P$ lies outside the ball and hence cannot be a hole point. If $|\vec{r}| < R_{max}$, then the point $P$ may be a hole point and a most precise test is needed. 6) Search for the point $P_c$, on $C$, which is closest to $P$. Compute $\vec{N}_p \cdot \vec{R}_p$ where $\vec{N}_p$ is the outward normal at the point $P_c$ on $C$ closest to $P$, and $\vec{R}_p$ is the position vector to $P$ from $P_c$. If $\vec{N}_p \cdot \vec{R}_p \geq 0$ , $P$ is outside of C, if $\vec{N}_p \cdot \vec{R}_p < 0$, $P$ is inside of $C$ and is a hole point.

* * *

Many of the problems encountered in constructing composite mesh schemes are encountered in other fields such as computer graphics and CAD/CAM. The area of computer science referred to as computational geometry (c.f. [31,32]) can be a source of ideas and algorithms.

## CONCLUDING REMARKS

For simple configurations finite difference solution procedures using well-ordered, body-conforming curvilinear grids have proven quite efficient for simulating the nonlinear, multiscale equations of aerodynamics. These same procedures can be used to simulate flow about complex configurations by incorporating them into composite grid schemes. Such a composite grid can be entirely comprised either of well-ordered grid elements or combinations of well-ordered grid elements and randomly ordered elements.

Although more complex than single grid schemes, composite grid approaches offer additional opportunities. For example, in composite mesh schemes it is somewhat natural to think of using simplified equations in isolated or specialized grids in which such approximations might be warranted. One possibility is to build a specialized Navier Stokes solver for a thin body conforming grid which is designed to resolve only the boundary layer — a sort of generalized boundary layer scheme.

In many ways the development of composite grids will change the field of computational aerodynamics. In the past one or two practitioners could build a more or less specialized code to treat all of the aspects of a particular nonlinear problem. Future composite grid codes will be larger, more complex, and perhaps menu driven from workstations. Hopefully they will be comprized of modular self-contained subcodes that can quickly be altered to accommodate improved numerical algorithms as they evolve.

## REFERENCES

1. Woodward, F. A. An Improved Method for the Aerodynamics Analysis of Wing -Body-Tail Configurations in Subsonic and Supersonic Flow. Part 1: Theory

and Applications. NASA CP 2228, Part 1, 1973.

2. Carmichael, R. L. and Erickson, L. L. PANAIR - A Higher Order Panel Method for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations. AIAA 81-1255, 1981.

3. Boppe, C. W. and Stern, M. A. Simulated Transonic Flows for Aircraft with Nacelles, Pylons, and Winglets. AIAA Paper No. 80-0130. Pasadena, CA. 1980.

4. Kafyeke, F. Prediction of Wing-Body-Store Aerodynamics Using a Small Perturbation Method and a Grid Embedding Technique. AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April, 1986.

5. Rai, M. M. Navier-Stokes Simulations of Rotor-Stator Interaction Using Patched and Overlaid Grids. AIAA Paper No. 85-1519, July, 1986.

6. Nakahashi, K. FDM-FEM Zonal Approach for Computations of Compressible Viscous Flows. 10th International Conference on Numerical Methods in Fluid Dynamics. Beijing, China, June, 1986.

7. Eberhardt, D. S., A Study of Multiple Grid Problems on Concurrent Processing Computers. Stanford University, Dept. of Aeronautics and Astronautics Thesis, Stanford, CA., Sept. 1984.

8. Hessenius, K.A.; and Rai, M.M.: Applications of a Conservative Zonal Scheme to Transient and Geometrically-Complex Problems. Computers and Fluids, Vol.14, No. 1, pp 43-58, June 1984.

9. Dougherty, F. C., Benek, J. A., and Steger, J. L. "On Application of Chimera Grid Schemes to Store Separation. NASA TM 88193, Oct. 1985.

10. Rai, M.M.: A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations. Jour. of Comp Physics, Vol. 62, No. 2, Feb. 1986.

11. Rai, M.M.: An Implicit Conservative Zonal Boundary Scheme for Euler Equation Calculations. AIAA 85-0488, January 1985.

12. Rubbert, P.E. and Lee, K.D., "Patched Coordinate Systems," Numerical Grid Generation, J.F. Thompson, ed., North-Holland, New York, N.Y., 1982, pp. 235-252.

13. Lee, K.D., "3-D Transonic Flow Computations Using Grid Systems with Block Structure," AIAA Paper No. 81-0998, June 1981.

14. Lasinski, T.A., Andrews, A.E., Sorenson, R.L., Chaussee, D.S., Pulliam, T.H., and Kutler, P., "Computation of the Steady Viscous Flow over a Tri- Element Augmentor Wing Airfoil,: AIAA Paper No. 82-0021, January 1982.

15. Flores, J., Holst, T. L., Kaynak, U., Gundy, K. and Thomas, S. D. Transonic Navier Stokes Wing Solution Using a Zonal Approach. Part 1. Solution Methodology and Code Validation. AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April, 1986.

16. Chaderjian, N. M. Transonic Navier Stokes Wing Solutions Using a Zonal Approach. Part 2. High Angle of Attack Simulation, AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April, 1986.

17. Karmen, S. L. Jr., Steinbrenner, J. P. and Kisielewski, K. M. Analysis of the F-16 Flow Field by a Block Grid Euler Approach, AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April, 1986.

18. Eberle, A. and Misegades, K. Euler Solution for a Complete Fighter Aircraft at Sub - and Supersonic Speed. AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April, 1986.

19. Rai, M.M.: A Relaxation Approach to Patched-Grid Calculations with the Euler Equations. AIAA Paper 85-0295, January 1985.

20. Magnus, R. and Yoshihara, H. Inviscid Transonic Flow over Airfoils. AIAA Paper 70 - 47, 1970.

21. Atta, E. H. Component Adaptive Grid Interfacing. AIAA Paper 81-0382, January 1981.

22. Atta, E. H. and Vadyak, J. A. : A Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows about Aircraft Configurations. AIAA Paper 82-1017, June 1982.

23. Steger, J.L., Dougherty, F.C., and Benek, J.A., "A Chimera Grid Scheme," Advances in Grid Generation, K.N. Ghia and U. Ghia, eds., ASME FED-Vol. 5, June 1983.

24. Benek, J.A., Steger, J.L., and Dougherty, F.C., "A Flexible Grid Embedding Technique with Application to the Euler Equations," AIAA Paper No. 83-1944, July 1983.

25. Lombard, C.K.; and Venkatapathy, E.: Implicit Boundary Treatment for Joined and Disjoint Patched Mesh Systems. AIAA Paper 85-1503-CP, July 1985.

26. Benek, J.A., Steger, J.L., Dougherty, F.C., and Buning, P. G., "Chimera: A Grid Embedding Technique" AEDC - TR - 85 - 64. April, 1986.

27. Berger, M.J., "Adaptive Mesh Refinements for Hyperbolic Partial Differential Equations," STAN-CS-82-924, Stanford University, August 1982.

28. Berger, M.J., "On Conservation at Grid Interfaces," ICASE Report 84-43, NASA Langley Research Center, Hampton, Virginia, September 1984.

29. Mastin, C.W. and McConnaughey, H.V., "Computational Problems on Composite Grids," AIAA Paper No. 84-1611, June 1984.

30. Benek, J. A., Buning, P.G., and Steger, J.L. A 3-D Chimera Grid Embedding Technique, AIAA Paper No. 1523, AIAA 7th Computational Fluid Dynamics Conference. Cincinnati, Ohio, July 15-17, 1985.

31. Preparata, F. P. and Shamos, M. I. Computational Geometry, An Introduction. Springer - Verlag, New York, 1985.

32. Brodlie, K. W. ed. Mathematical Methods in Computer Graphics and Design. Academic Press, London, 1980.

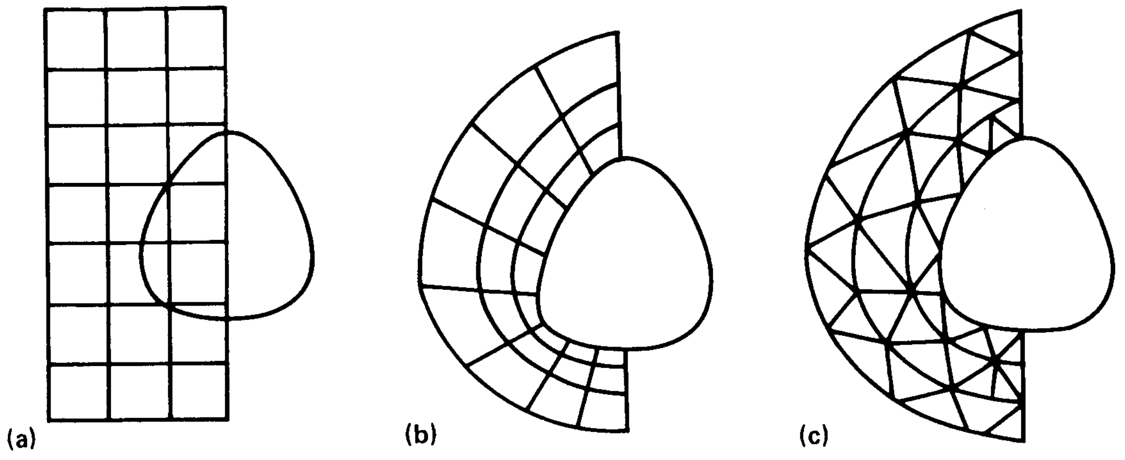


Figure 1. Sketches showing basic grid treatments for a simple body, a. rectangular or Cartesian, b. body conforming curvilinear, c. body conforming irregular triangularized.

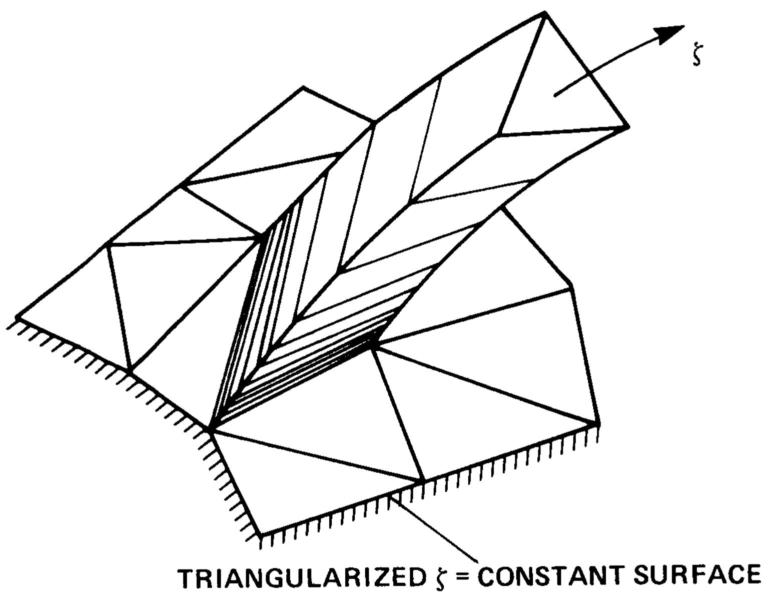


Figure 2. Sketch showing semi-ordered "prismatic" grid in which $\varsigma$ varies along outward radiating lines and each $\varsigma = constant$ plane is triangularized.

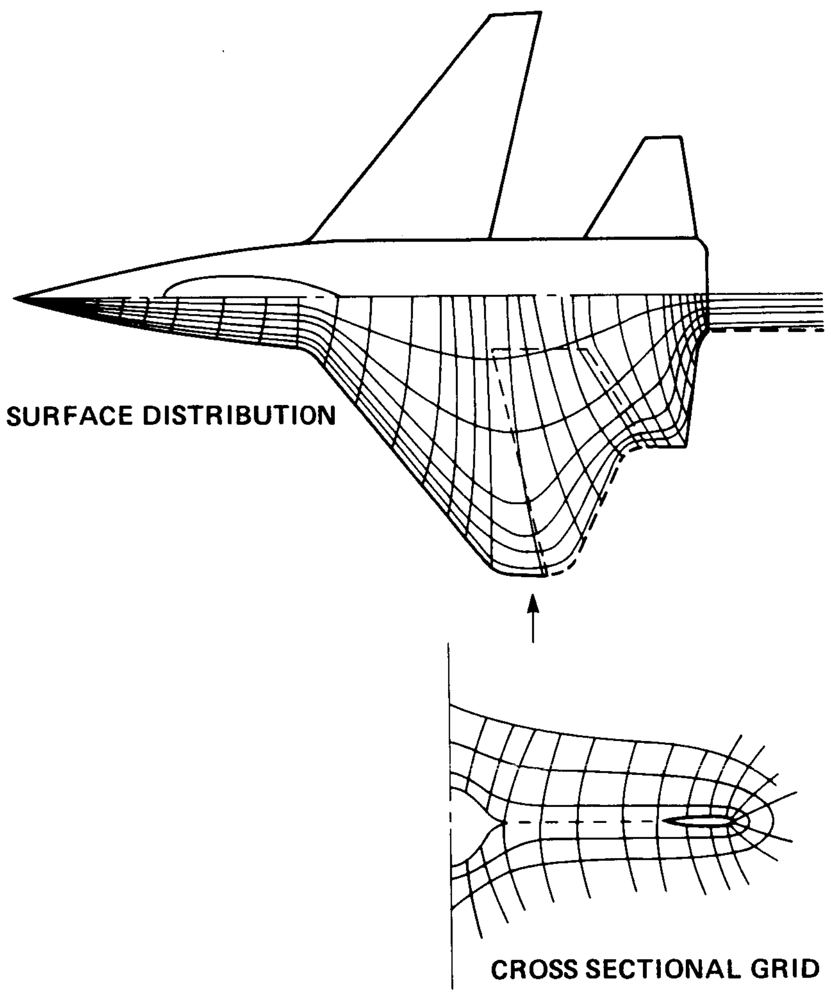


Figure 3. Sketch illustrating the use of a single body conforming grid with cuts or webs inserted to give a single body surface.

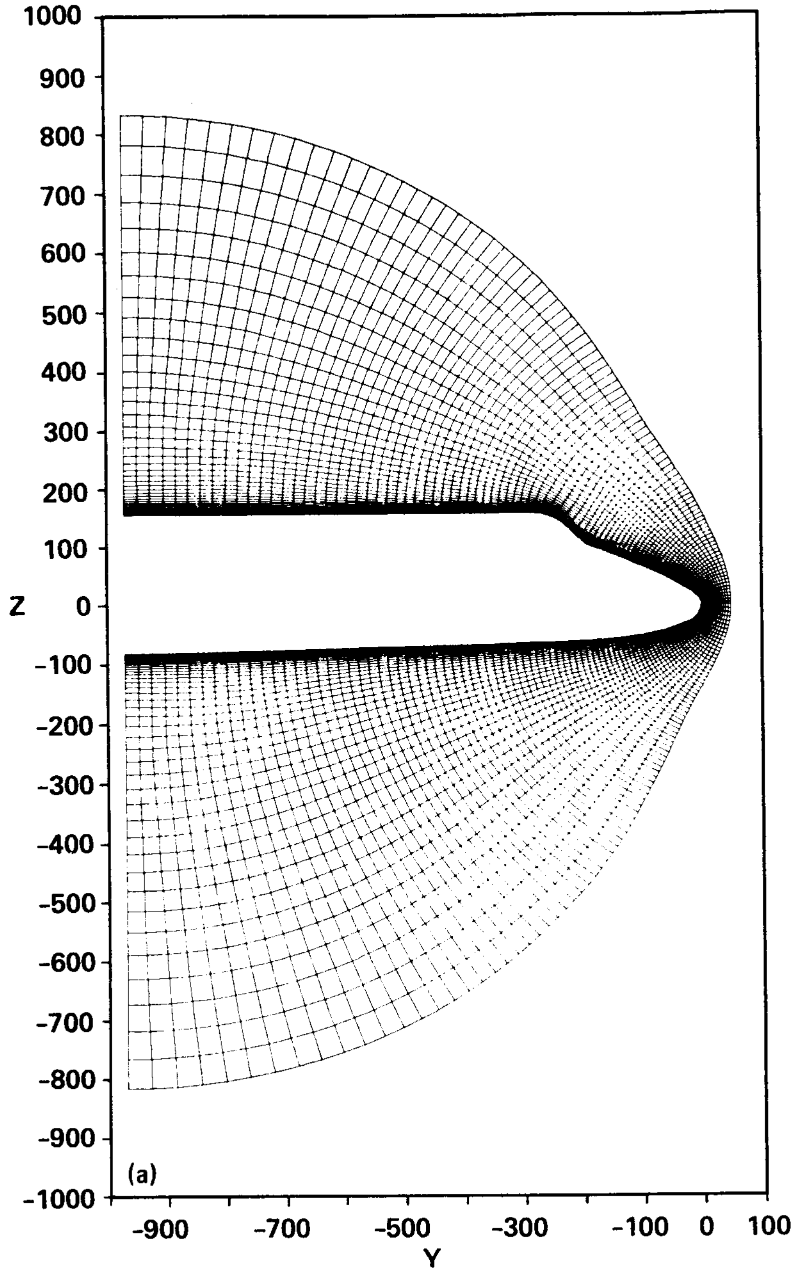


Figure 4. Use of a single body conforming mesh about the shuttle orbiter using an outflow boundary condition. a. Windward and leeward plane grid segments.

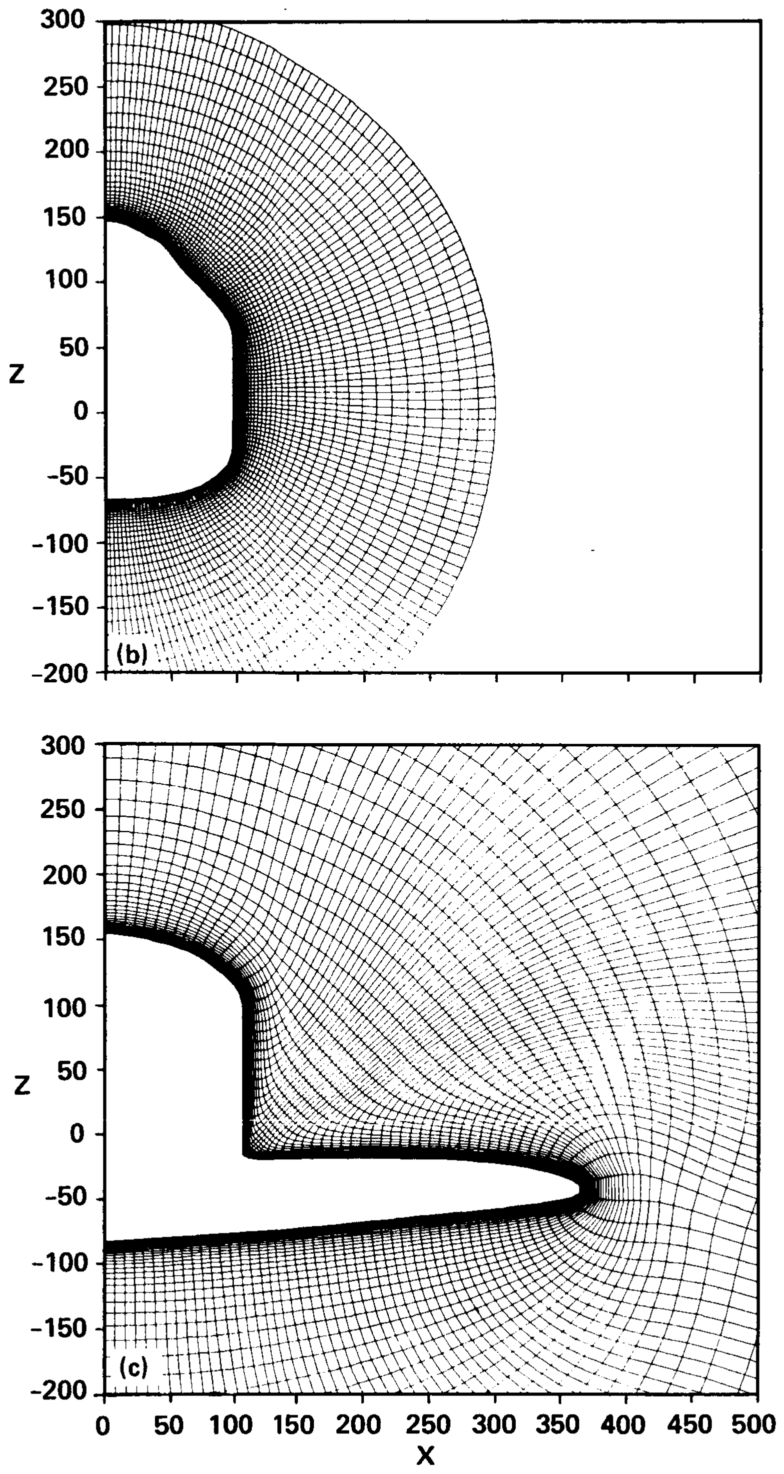


Figure 4 continued. Cross sectional views of curvilinear grid projected onto an $x = constant$ plane. b. Near canopy location. c. Wing region.

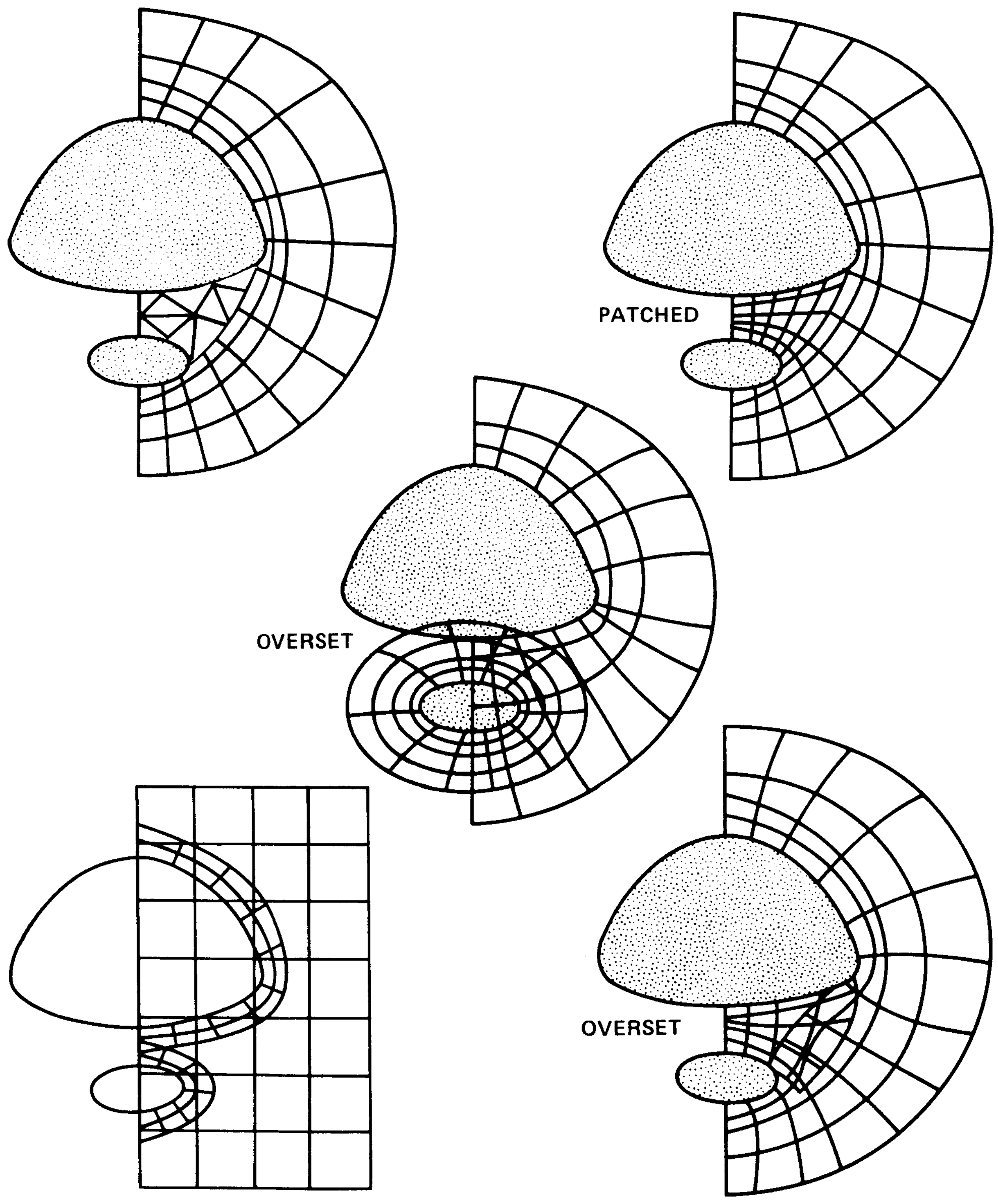
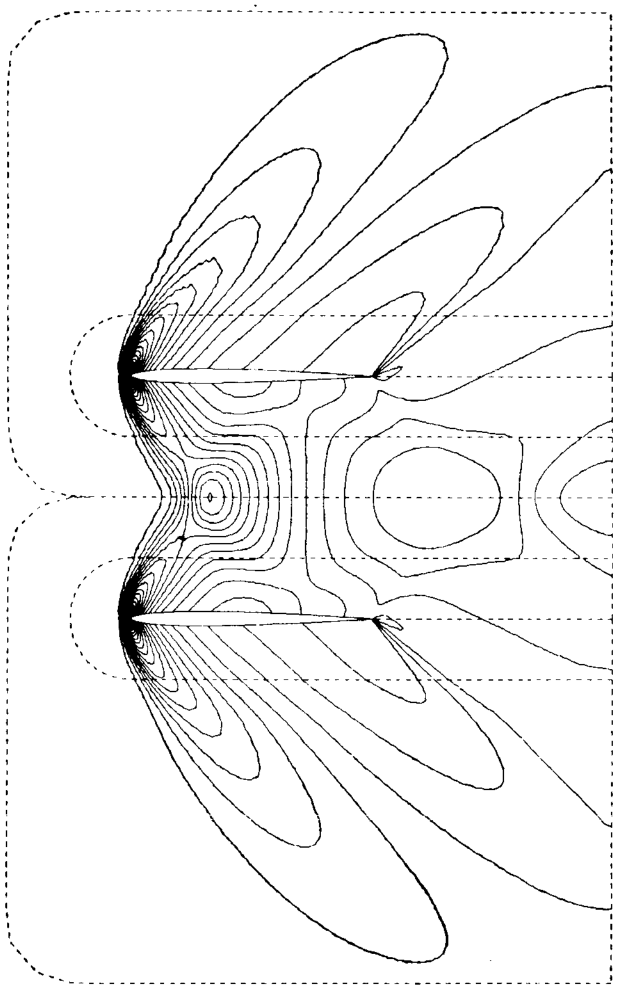


Figure 5. Possible composite grid treatments of a multiple body combination using patched and overset grids.



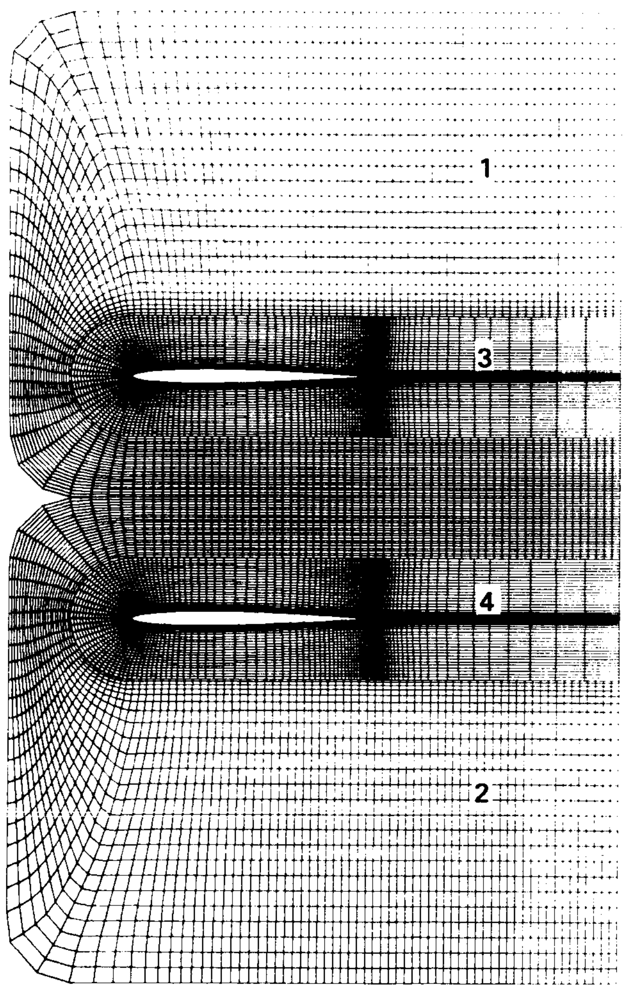


Figure 6. Inviscid supersonic flow about a biplane computed using patched grids, $M_\infty = 1.5$. a. Mach contours. b. Composite grid formed using four patches.

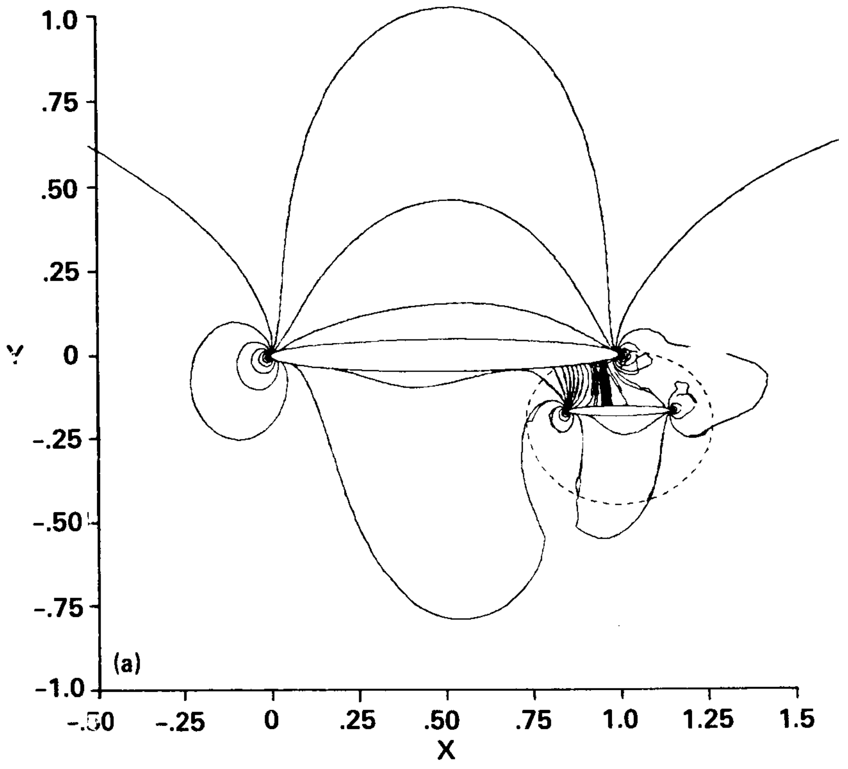
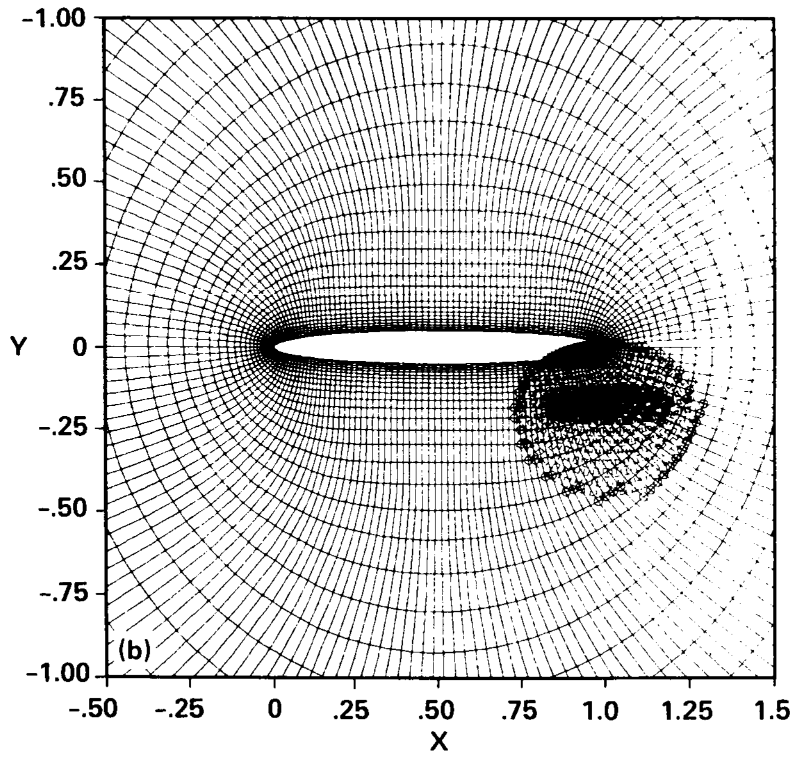


Figure 7. Inviscid transonic flow about an airfoil flap combination using overset grids, $M_\infty = 0.7$. a. Mach contours.



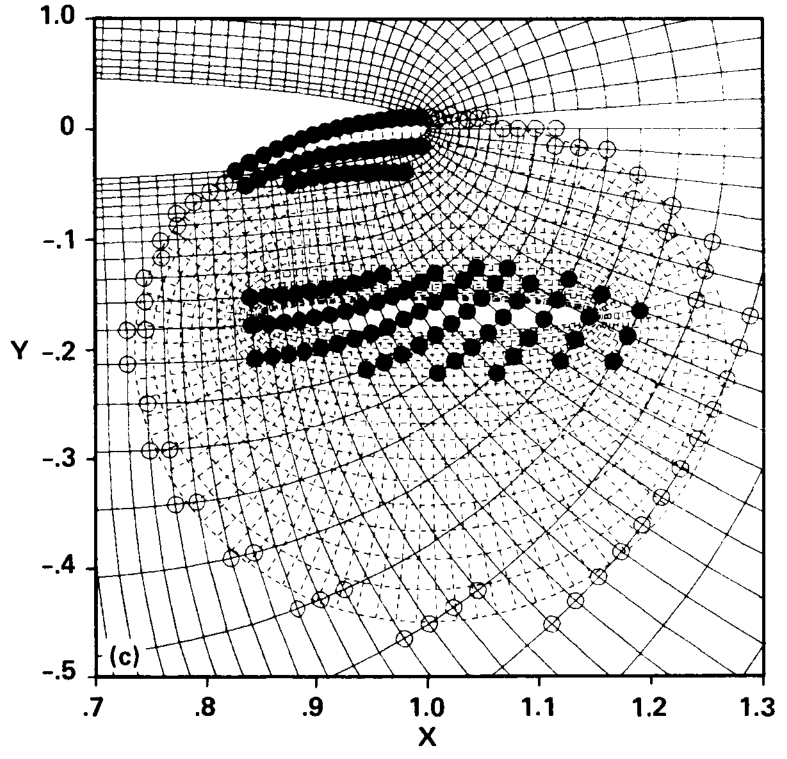


Figure 7 continued. b. Composite grid formed using two overset grids. c. Detail of minor grid showing points blanked out (filled ○) and points used to interpolate the outer boundary (open ○).

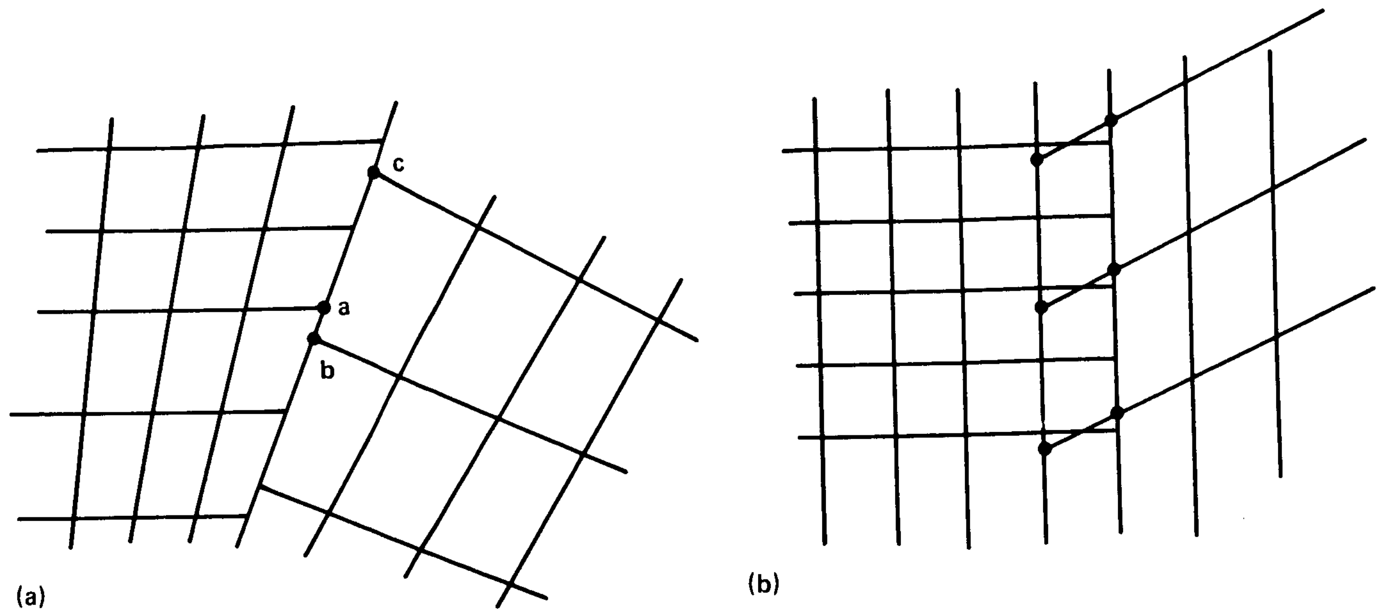


Figure 8. Sketches showing the interface boundary of discontinuous grids.

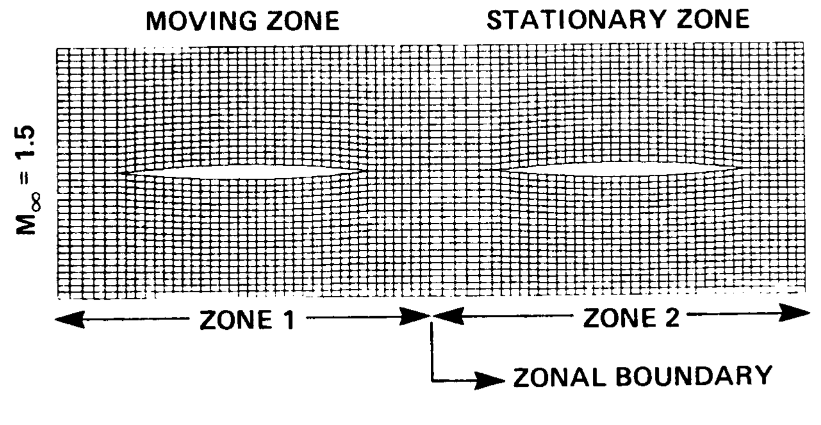


Figure 9. Two-zone patched grid for rotor-stator simulation.

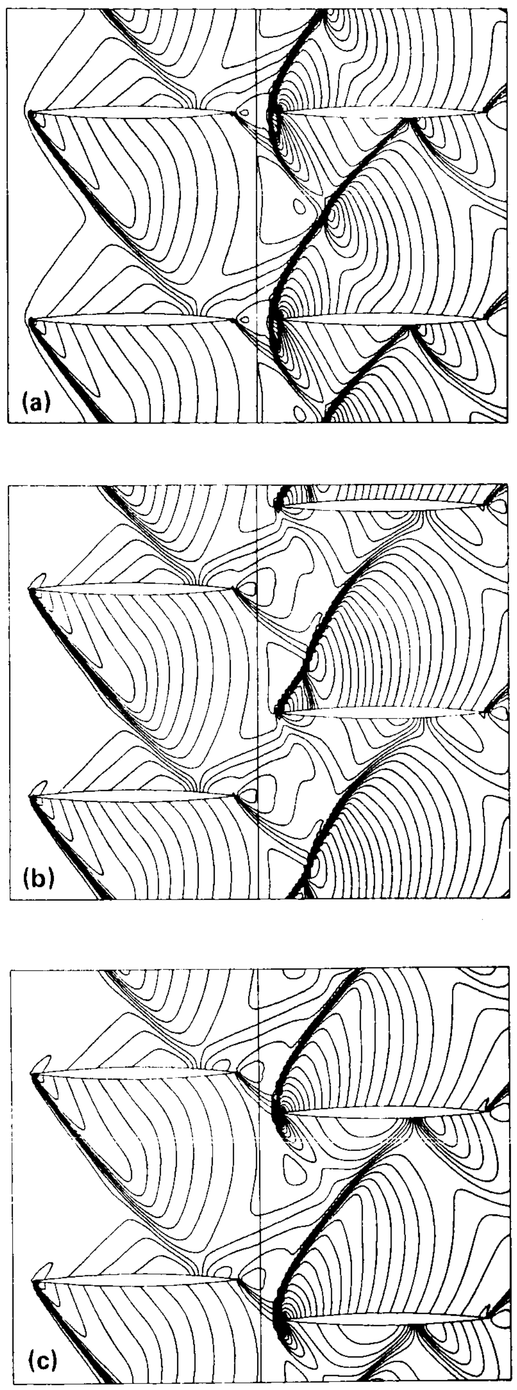


Figure 10. Pressure contours of rotor-stator simulation at various times.

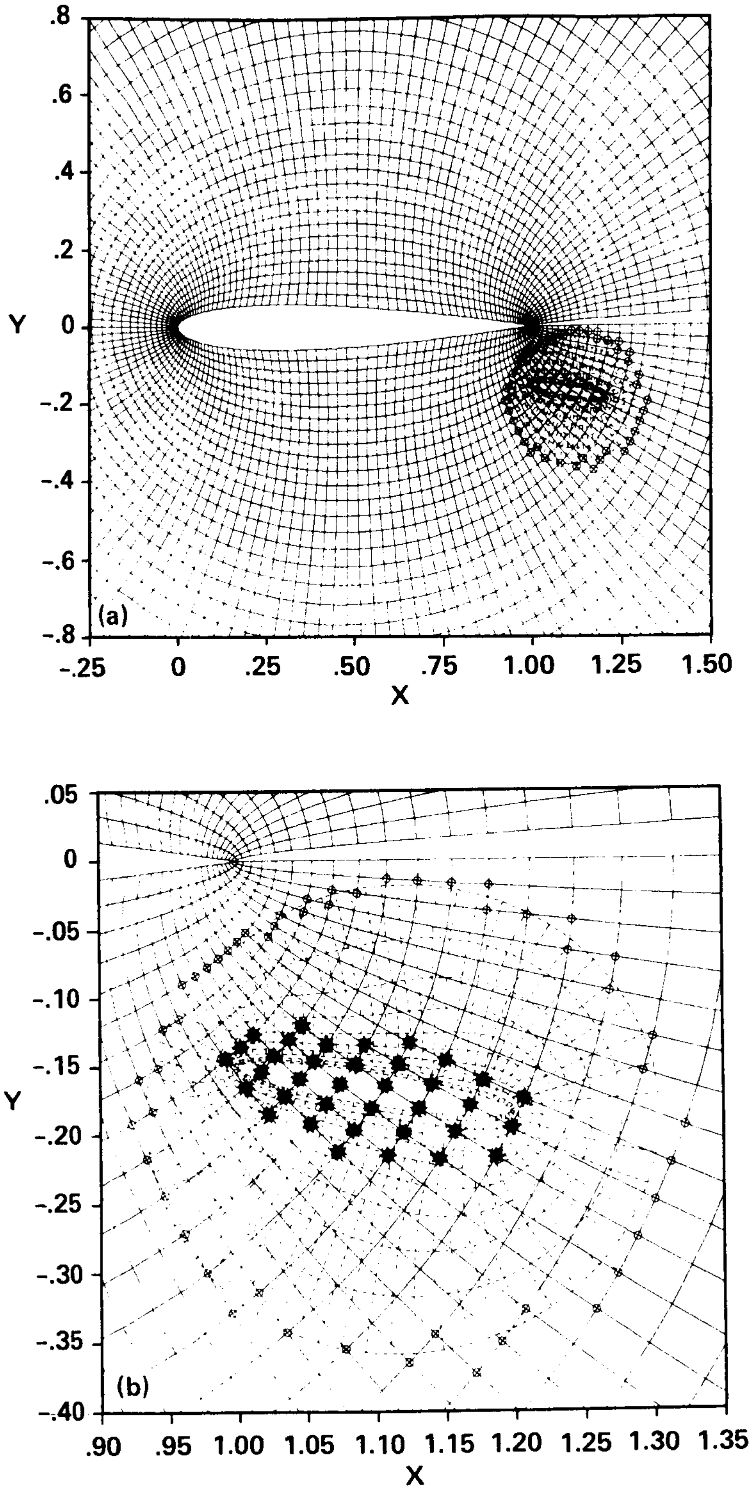


Figure 11. Overset grids for simple airfoil-flap arrangement.

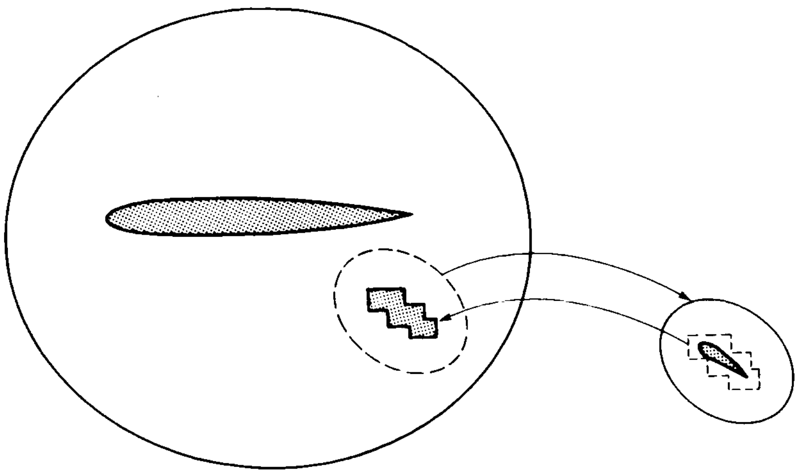


Figure 12. Sketch illustrating transfer of information between overset grids.

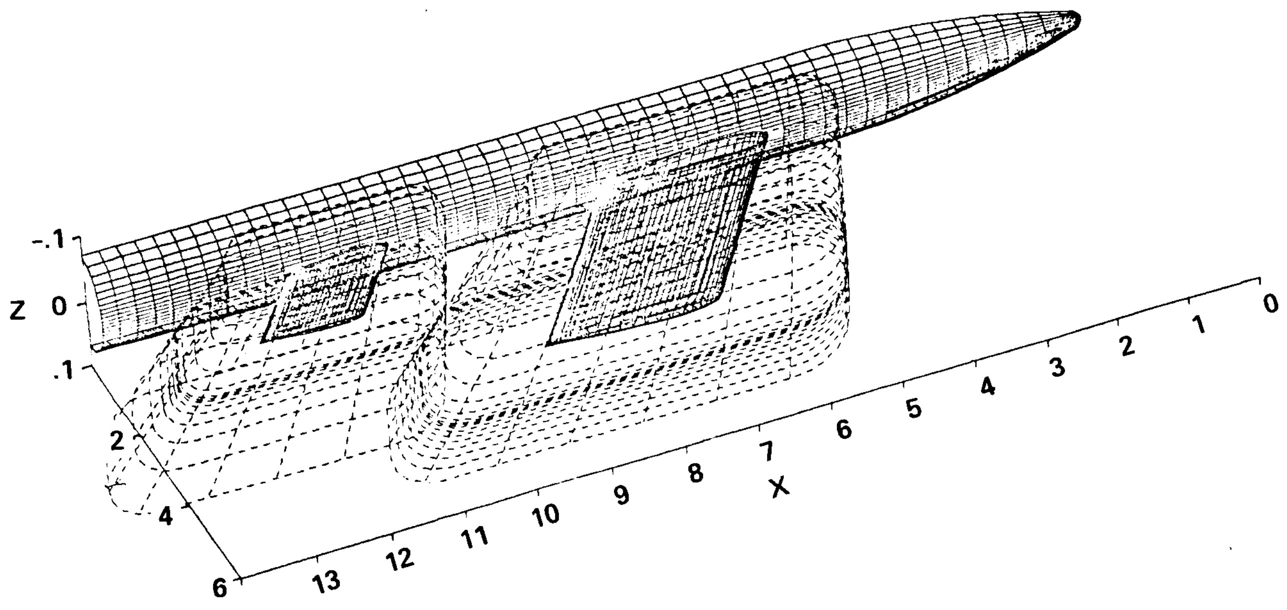


Figure 13. Overset grids and pressure distributions for a generic wing-body-tail configuration, $M_\infty = 0.9$ and $\alpha = 2°$. a. Surface mesh and wing and tail outer boundary surfaces.

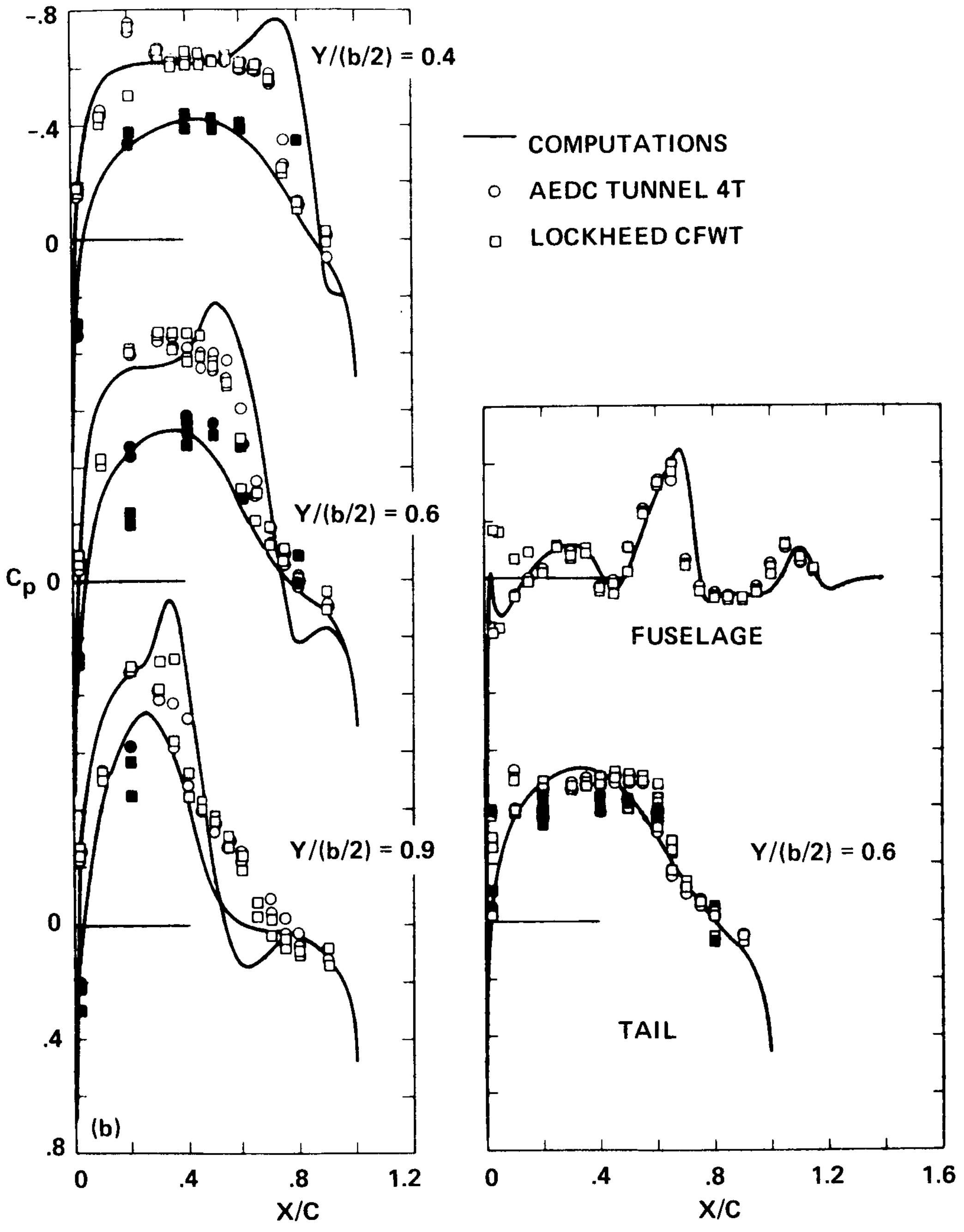
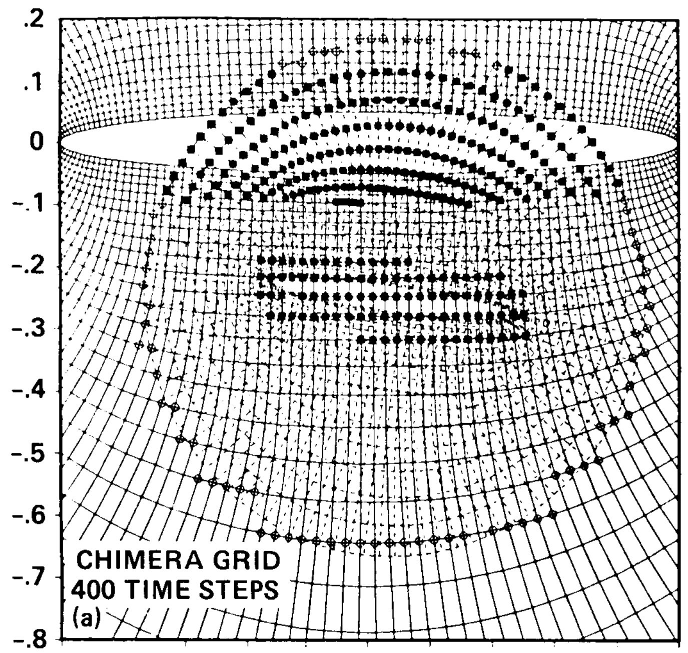


Figure 13 cont. b. Computed and experimental pressure distributions.



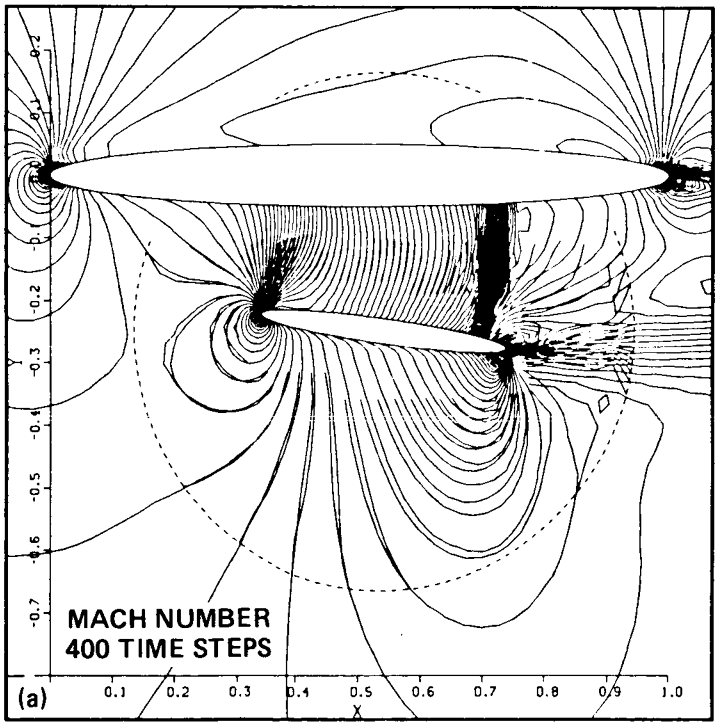
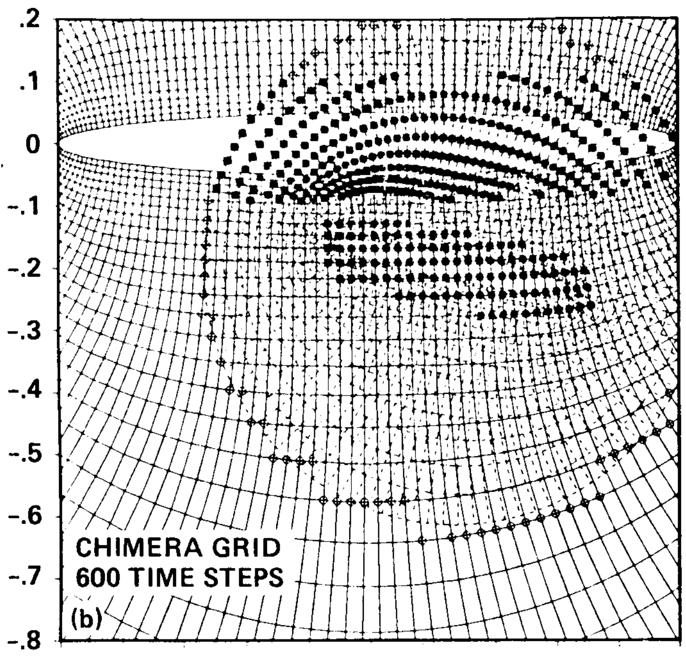


Figure 14. Overset grids and Mach contours at different times for an unsteady two-body separation, $M_\infty = 0.7$. a. Time of 400 units.



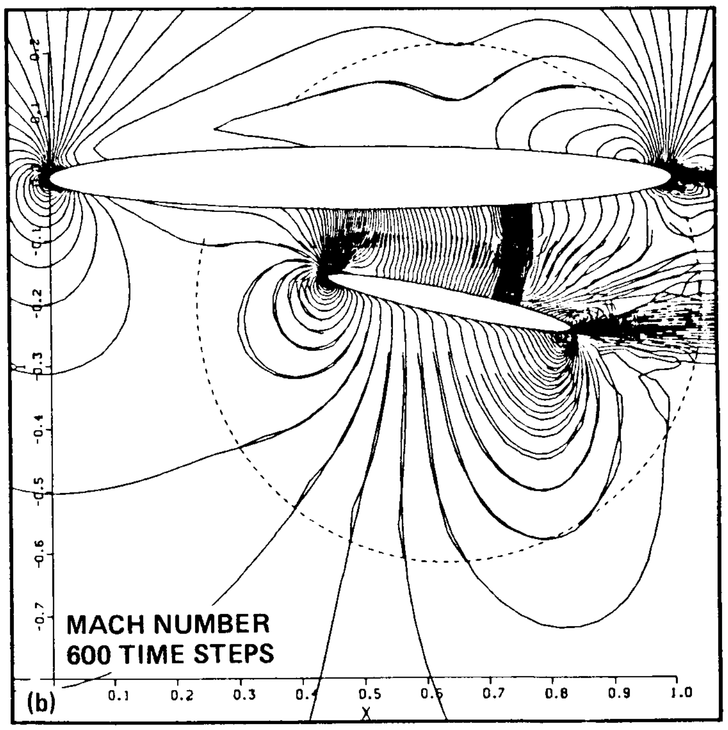


Figure 14 continued. b. Time of 600 units.

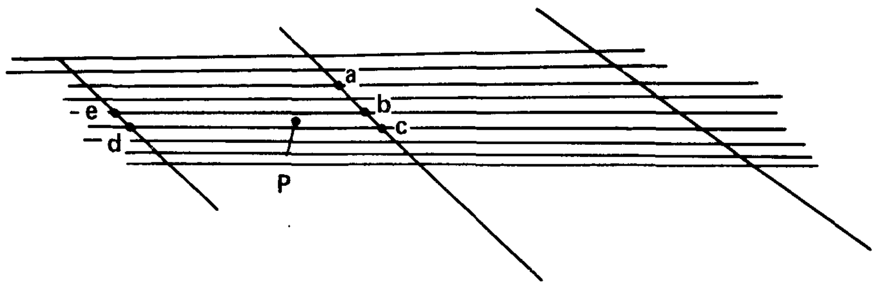


Figure 15. Sketch showing situation in which nearest point, $a$, is not the best point to use for interpolation of point $P$.

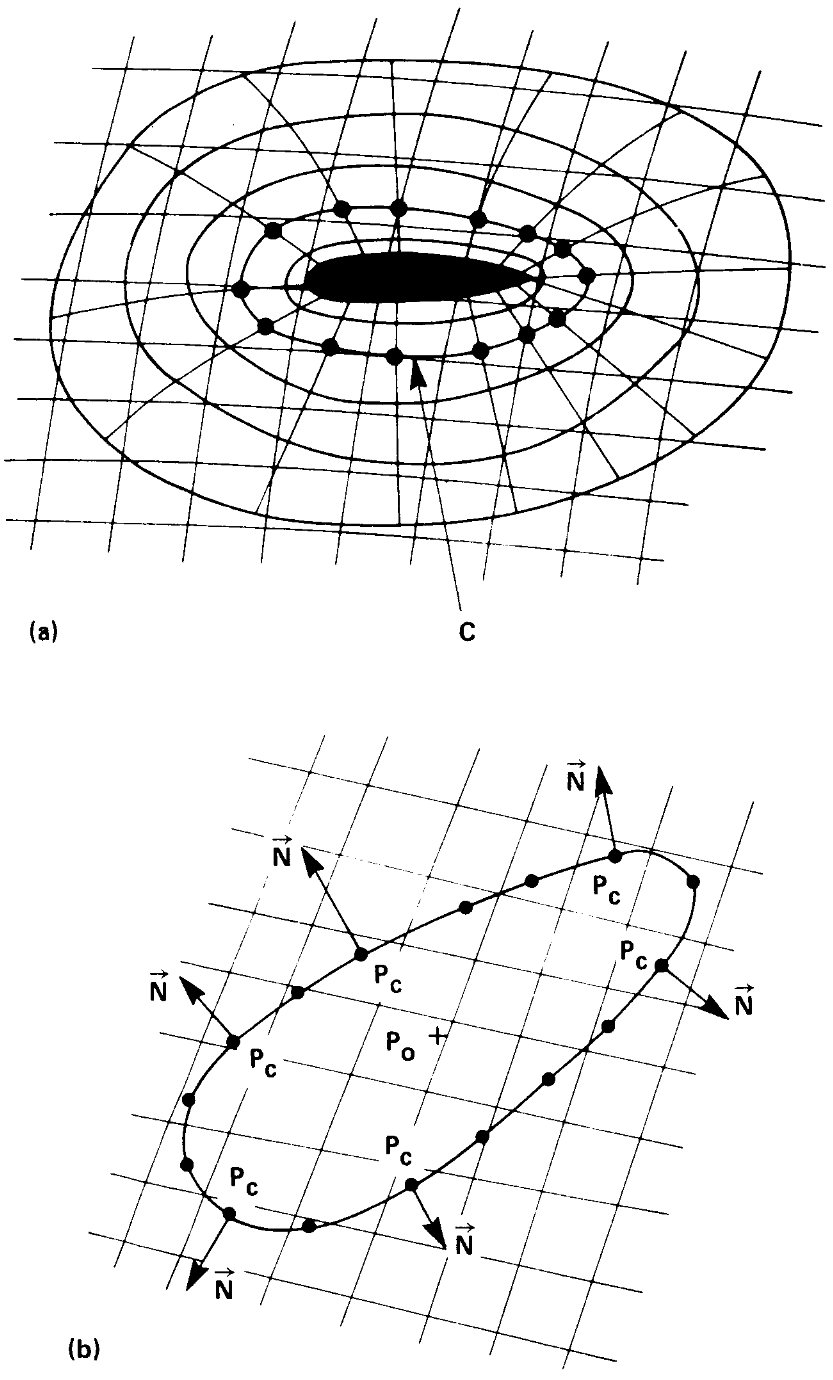
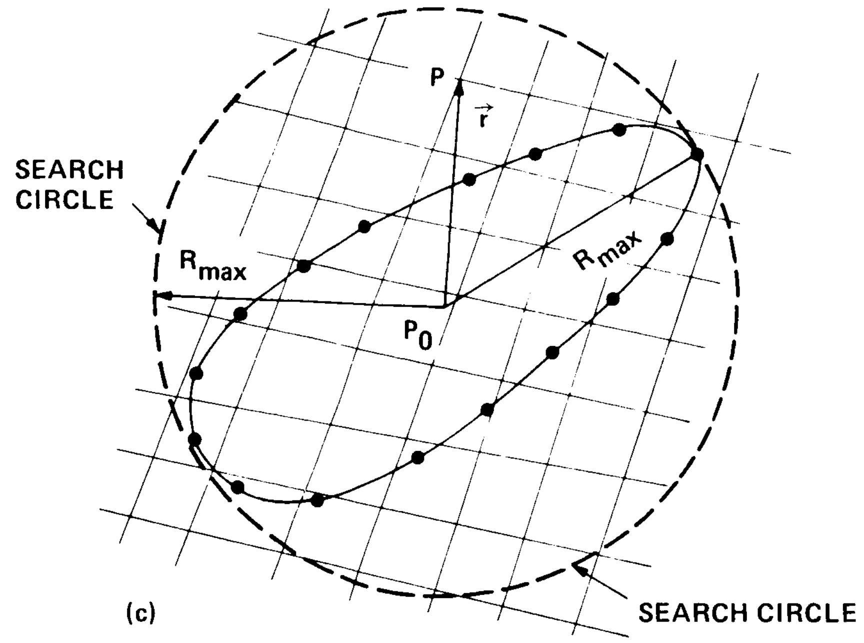


Figure 16. Hole construction in two dimensions. a. Initial hole boundary defined by level curve, C. b. Constructions of outward normals to curve, C.



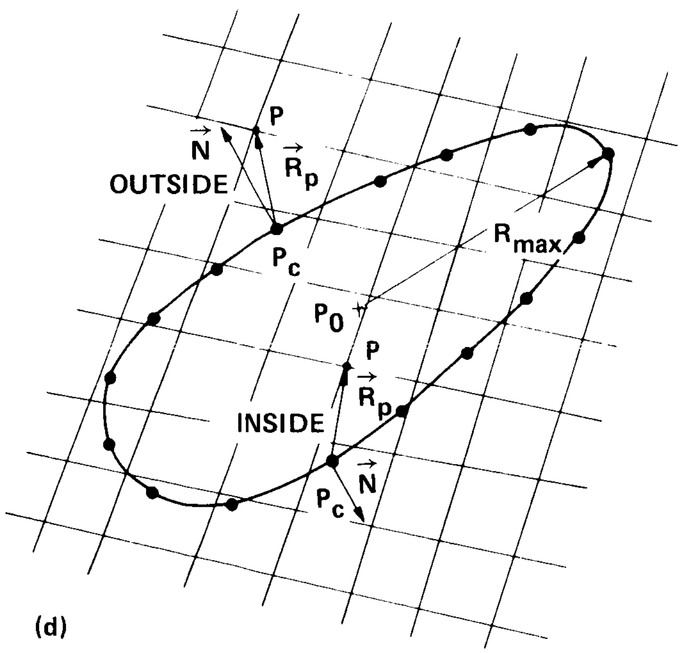


Figure 16 continued. c. Temporary origin and construction of search ball or circle. d. Construction of position vector, $\vec{R}_p$, and dot product test.

| | | |
|---|---|---|
| 1. Report No. NASA TM-88372 | 2. Government Accession No. | 3. Recipient's Catalog No. |
| 4. Title and Subtitle ON THE USE OF COMPOSITE GRID SCHEMES IN COMPUTATIONAL AERODYNAMICS | | 5. Report Date November 1986 |
| | | 6. Performing Organization Code |
| 7. Author(s) Joseph L. Steger and John A. Benek (Calspan Corp., AEDC, Arnold AFS, Tennessee) | | 8. Performing Organization Report No. A-86425 |
| | | 10. Work Unit No. |
| 9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035 | | 11. Contract or Grant No. |
| | | 13. Type of Report and Period Covered Technical Memorandum |
| 12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546 | | 14. Sponsoring Agency Code 505-60 |

15. Supplementary Notes

Point of Contact: Joseph L. Steger, Ames Research Center, MS 202A-1, Moffett Field, CA 94035
(415)694-6459 or FTS 464-6459

16. Abstract

In finite difference flow field simulations, the use of a single well-ordered body conforming curvilinear mesh can lead to efficient solution procedures. However, it is generally impractical to build a single grid of this type for complex three-dimensional aircraft configurations. As a result, a trend in computational aerodynamics has been toward the use of composite grids. Composite grids use more than one grid to mesh an overall configuration with each individual subgrid of the system patched or overset together. Because each individual subgrid in the system is well-ordered, the overall grid is suitable for efficient finite difference solution using vectorized or multitasking computers. Some of the advantages and difficulties of using various composite grid schemes are reviewed in this paper.

| | |
|---|---|
| 17. Key Words (Suggested by Author(s)) CFD, Composite grids, Computational fluid dynamics, Finite difference methods | 18. Distribution Statement Unclassified – Unlimited; Subject Category – 02 |

| | | | |
|---|---|---|---|
| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price* |
| Unclassified | Unclassified | 32 | A03 |

*For sale by the National Technical Information Service, Springfield, Virginia 22161